AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

**WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI,
INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ**

KATEDRA INFORMATYKI STOSOWANEJ

Praca dyplomowa inżynierska

Implementacja systemu uwierzytelniania z zastosowaniem Negatywnych Baz Danych

Implementation of authentication system using Negative Databases

Autor: *Grzegorz Nieużyła*
Kierunek studiów: *Informatyka*
Opiekun pracy: *dr inż. Piotr Szwed*

Kraków, 2020

Uprzedzony o odpowiedzialności karnej na podstawie art. 115 ust. 1 i 2 ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.): „Kto przywłaszcza sobie autorstwo albo wprowadza w błąd co do autorstwa całości lub części cudzego utworu albo artystycznego wykonania, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 3. Tej samej karze podlega, kto rozpowszechnia bez podania nazwiska lub pseudonimu twórcy cudzy utwór w wersji oryginalnej albo w postaci opracowania, artystycznego wykonania albo publicznie zniekształca taki utwór, artystyczne wykonanie, fonogram, wideogram lub nadanie.", a także uprzedzony o odpowiedzialności dyscyplinarnej na podstawie art. 211 ust. 1 ustawy z dnia 27 lipca 2005 r. Prawo o szkolnictwie wyższym (t.j. Dz. U. z 2012 r. poz. 572, z późn. zm.): „Za naruszenie przepisów obowiązujących w uczelni oraz za czyny uchybiające godności studenta student ponosi odpowiedzialność dyscyplinarną przed komisją dyscyplinarną albo przed sądem koleżeńskim samorządu studenckiego, zwanym dalej «sądem koleżeńskim».", oświadczam, że niniejszą pracę dyplomową wykonałem(-am) osobiście i samodzielnie i że nie korzystałem(-am) ze źródeł innych niż wymienione w pracy.

Spis treści

1. Wprowadzenie

1.1. Cele pracy

Celem niniejszej pracy jest implementacja i przetestowanie systemu uwierzytelniania oferującego większe bezpieczeństwo niż standardowy schemat generowania skrótu hasła za pomocą funkcji generacji klucza (np. PBKDF2, bcrypt) i przechowywaniu w standardowej (pozytywnej) bazie danych.

Założeniem systemu jest zamienienie reprezentacji w sposób jawny na Negatywną Bazę Danych (NDB) co pozwoli dodać dodatkową warstwę bezpieczeństwa która znacząco utrudni uzyskanie haseł użytkownika w przypadku wykradzenia bazy danych.

Rezultatem wyjściowym algorytmów generacji NDB jest zbiór formuł logicznych, dlatego przeprowadzone zostały testy z wykorzystaniem solwerów SAT mające na celu zasymulowanie ataku na powstałą NDB.

1.2. Zawartość pracy

2. Negatywne Bazy Danych - opis teoretyczny

2.1. Opis działania

Główną operacja wykonywalną na NDB jest sprawdzenie czy dany rekord znajduje się w bazie. Przyjmując U jako oznaczenie uniwersum języka binarnego o długości l a DB jako zbiór wszystkich rekordów, każdy o długości l, NDB przechowuje zbiór $U - DB$ [1]. Takie dane są niepraktycznie do zareprezentowania w postaci nieskompresowanej z uwagi na wielkość, dlatego stosuje się wyrazy nad alfabetem $\{0, 1, *\}$ gdzie symbol $*$ może oznaczać zarówno 0 lub 1 w jawnej reprezentacji bitowej. Każdy taki wyraz odpowiada jednemu lub wielu elementom $U - DB$ i jest sprowadzany do formuły logicznej (Tabela 2.1). Z założenia algorytm sprawdzający przynależność do DB sprawdza czy jakakolwiek formuła z NDB jest spełniana przez dany rekord. Dane znajdują się w DB wtedy i tylko wtedy gdy żadna formuła nie zostanie spełniona. Taki model działania wymusza na danych stałą wielkość, co jednak nie stanowi problemu w przypadku przechowywania skrótów haseł które mają stałą, zależną od konkretnego algorytmu długość.

Tabela 2.1. Reprezentacja formuł logicznych za pomocą NDB

rekord NDB	formuła logiczna
011*	$\neg x_1 \wedge x_2 \wedge x_3$
0*01	$\neg x_1 \wedge \neg x_3 \wedge x_4$
111*	$x_1 \wedge x_2 \wedge x3$

2.2. Zastosowanie w systemach uwierzytelniania

NDB może być wykorzystana w każdym systemie, gdzie podstawową operacją na danych jest sprawdzenie czy dany rekord znajduje się w bazie. Jednym z najpopularniejszych systemów uwierzytelniania jest metoda oparta na loginie i haśle. Użytkownik danej aplikacji przy zakładaniu konta podaje hasło, które następnie warstwa serwerowa danej aplikacji przechowuje jako wynik nieodwracalnej funkcji haszującej.

W przypadku nieautoryzowanego dostępu do bazy danych i używanego algorytmu uzyskiwania skrótu z hasła, atakujący może uzyskać wartość pierwotną mało skomplikowanych haseł za pomocą

np. metody słownikowej. Modyfikując powyższy algorytm składując skróty jako rekordy w NDB uniemożliwiamy iterację wszystkich danych, jednocześnie pozostawiając łatwy dostęp do informacji czy użytkownik o podanym loginie i haśle ma dostęp do aplikacji.

2.3. Algorytmy generowania Negatywnych Baz Danych

2.3.1. Algorytm prefiksowy

Najprostszym ze sposobów generowania Negatywnych Baz Danych jest zaproponowany przez Fernando Esponda algorytm prefiksowy [1, 2]. Został on opracowany w celu udowodnienia że proces generowania NDB z rekordów DB jest możliwy w rozsądnej złożoności czasowej i pamięciowej.

Algorithm 1: Algorytm prefiksowy

Data: DB - zbiór rekordów do zareprezentowania w NDB, l - liczba rekordów w DB

Result: Zbiór rekordów NDB

$Prefix_n(V)$ - Prefiks n-znakowy rekordu V

$len(V)$ - Długość rekordu V

W_i = {};

i = 0;

while $i < l$ **do**

  W_i = { V_p | $len(V_p)$ = i + 1, $V_p \notin \{Prefix_{i+1}(V)|V \in DB\}$, $Prefix_i(V_p) \in W_i$}

  for V_p *in* W_i **do**

    Stwórz rekord NDB o długości l którego V_p jest prefiksem a na pozostałych pozycjach jest symbol $*$ i dodaj do zbioru wyjściowego

  end

  i = i + 1;

  W_i = { $Prefix_i(V)|V \in DB$}

end

Opis działania. *Powyższa metoda polega na generowaniu coraz dłuższych prefiksów które nie pokrywają się ze zbiorem DB. W ten sposób na początku tworzone są rekordy odpowiadające znacznej części $U - DB$. Czasami występuje potrzeba zdefiniowania pewnych rekordów explicite bez wykorzystania symbolu $*$ jeżeli każdy możliwy prefiks jest także prefiksem rekordu z DB. Przykładowy wynik działania znajduje się w tabeli 2.2.*

Algorytm prefiksowy jest deterministyczny i każdy powstały rekord reprezentuje unikalną, nie pokrywającą się część $U - DB$ [1]. Powoduje to, że algorytm uzyskiwania zbioru DB z otrzymanej

DB	U - DB	NDB
0000	0001	10**
0110	0011	010*
0010	0100	111*
1101	0101	0001
	0111	0011
	1000	0111
	1001	1100
	1010	
	1011	
	1100	
	1110	
	1111	

Tabela 2.2. Rezultat działania algorytmu prefiksowego

NDB nie wymaga sprowadzenia do problemu SAT. Wystarczy jedynie odpowiednio posortować rekordy i wyznaczyć przedziały pomiędzy nimi.

2.3.2. Algorytmy generujące trudne do odwrócenia Negatywne Bazy danych

3. Solwery SAT

3.1. Opis działania

3.2. Wykorzystywanie solwerów SAT w celu uzyskania przeciwobrazu Negatywnej Bazy Danych

4. Implementacja systemu uwierzytelniania

4.1. Reprezentacja danych

4.2. Algorytm tworzenia użytkownika

4.3. Algorytm uwierzytelniania użytkownika

4.4. Działanie aplikacji

5. Testy implementacji

5.1. Opis testowania za pomocą solwerów SAT

5.2. Testy algorytmów prostych

5.3. Testy algorytmów złożonych

6. Wnioski

6.1. Bezpieczeństwo przedstawionej implementacji

6.2. Możliwe rozszerzenia

Bibliografia

[1] Fernando Esponda. “Negative Representations of Information”. PhD thesis. 2005.

[2] F. Esponda, S. Forrest i P. Helman. „Enhancing Privacy through Negative Representations of Data”. W: 2004.